UNCLASSIFIED

AD 221 220

1 OF 1

FOR
MICRO-CARD
CONTROL ONLY

UNCLASSIFIED

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

Reprint No. 29

Nonlinear Network Flows and Convex Programming over Incidence Matrices

by

Abraham Charnes and William W. Cooper

1959

CARNEGIE INSTITUTE OF TECHNOLOGY

Pittsburgh 13, Pennsylvania

NONLINEAR NETWORK FLOWS AND CONVEX PROGRAMMING OVER INCIDENCE MATRICES*

A. Charnes†

Purdue University and Northwestern University

and

W. W. Cooper†

Carnegie Institute of Technology

INTRODUCTION

Problems involving network analysis arise in many industrial contexts. For example, interest often attaches to methods for characterizing the behavior of firewater safety systems in refinery operations. Flows and resulting pressures need to be ascertained under various patterns of input and withdrawal. Under certain conditions standard approaches such as Hardy Cross[1] methods and extensions of the usual Gauss-Seidel techniques, have been found difficult or impossible to apply ~~in analyzing~~ such systems. The nonlinear character of the branch head losses may cause lack of convergence of the solutions or require recourse to supplementary analyses of a complicated character.

Certain kinds of network problems have proved amenable to the usual formulations of linear programming.[2] ~~Here it is proposed to utilize~~ a ~~somewhat~~ different approach in order to provide a more general method of attack and ~~to do so, moreover, in a way which provides~~ access to the methods ~~described in [4]~~ for treating nonlinear problems as well. The essential ideas are as follows: (1) restatement of the original problem in terms of minimizing a separable convex[3] functional subject to a system of linear equations expressing the conservation of current entering and leaving nodes by branches;[4] (2) utilization of piecewise linear functionals to secure suitable approximations to the convex functionals; and (3) using the "bounded variables techniques"[5] of linear programming in order to avoid expanding the number of vectors in an

*Manuscript received December 1957.

†The authors are indebted to W. P. Drews of Esso Research and Engineering Co. and Paul Harthorn and Norman J. Driebeck of Creole and Lago Petroleum Cos., respectively, for background and advice.

The research underlying this paper was undertaken for the project Methodological Aspects of Management Research under contract with The Office of Naval Research; Contract Nonr-1100(05), Project NR 047 016; report issued August 1957.

[1]See [1].

[2]See, e.g., W. Prager, [9] and the references cited therein.

[3]By "separable convex" function is meant one which is the sum of convex functions of one variable each.

[4]I.e., it is assumed that leakages, if any, may be ignored.

[5]Cf. [8].

+ [Contract Nonr - 760(01)]

"active" basis and also to avoid expanding the number of different coefficient vectors which require explicit expression. The resulting problem is thereby converted to an equivalent one which may be called the polygonal approximation to convex programming over the "incidence matrix" of a connected network. Special properties are then secured which simplify analysis and representation, and provide a basis for systematic elaboration.

INCIDENCE MATRIX

In engineering analyses of networks, it is customary to designate one of the two possible directions of flow in a branch as positive. The current is then allowed to have either positive or negative values. Positivity and negativity are then respectively associated with flow in the positive direction or its opposite.

Topological analysis utilizes instead the closely related notion of "incidence" of branches (or links) on nodes. Thus, the convention may be adopted, for example, that a branch entering a node, is positively incident on that node, a branch leaving a node is negatively incident on it, and a branch which does not meet a node is not incident on it. The incidences of branches on nodes may then be described by an incidence matrix consisting of numbers ϵ_{ij} representing the incidence of the j^{th} branch on the i^{th} node as follows:

$$(1) \qquad \epsilon_{ij} = \begin{Bmatrix} 1 \\ -1 \\ 0 \end{Bmatrix} \begin{array}{l} \text{if branch } j \text{ is positively incident on node } i. \\ \text{" " } j \text{ " negatively " " " } i. \\ \text{" " } j \text{ " not " " " } i. \end{array}$$

Since each branch is incident on precisely two nodes each column of the matrix will contain precisely two nonzero entries—a plus one and a minus one. If q_j designates the current in the j^{th} branch the net influx of current into the i^{th} node is evidently

$$(2) \qquad \sum_j \epsilon_{ij} q_j ,$$

where j ranges over all branches of the network. If the efflux out of the network at this node is E_i then this conservation of current at nodes (Kirchhoff's node law) may be written as

$$(3) \qquad \sum_j \epsilon_{ij} q_j = E_i, \quad i = 1, 2, \ldots$$

Thus, to express the conservation conditions, it is necessary only to form the incidence matrix column by column.

BASES OF COLUMN VECTORS FOR AN INCIDENCE MATRIX

As stated in (1) above, each column vector of such a matrix contains a plus one and a minus one and is uniquely associated with a branch of the network. The plus one corresponds to the node at the head of the branch and the minus one corresponds to the node at the tail of the branch. A basis of column vectors of the matrix is a collection of column vectors (or branches) such that, (1) every other column vector can be expressed as a linear combination of this collection, and (2) the number of vectors in this collection is the least with which this task can be accomplished. Clearly, a basis must include at least one vector incident (positively or negatively) on each node of the system. Otherwise, it would not be possible to express (in terms of the basis) a vector associated with a branch incident on that node. To express a column vector (branch or node pair) in terms of the basis, it is necessary only to start with the head node of this branch and trace a path along branches associated with the basis to the tail node of this branch. The corresponding algebraic process consists of adding or subtracting the basis vectors associated with the branches thus traversed.

Consider, for example, the process of moving from one branch to another across a node. Both branches are incident on this node. If both have the same incidence, subtracting them will give a column vector with precisely one "plus one" and one "minus one" associated with the two end (or exterior) nodes of these two branches; if the branches are oppositely incident on a node, an addition will be performed to attain a vector corresponding to the end nodes. Thus, traversal of the path from node to node is associated with addition and subtraction of vectors so that as any particular node in the traversal is reached the cumulant vector has nonzero entries (of opposite sign and unit magnitude) corresponding precisely to the end nodes of the completed portion of the path.

The collection of branches corresponding to the basis can contain no closed path (or cycle). Otherwise it would be possible to express one of the branches in the cycle in terms of the others by the process just described. Every expression containing the vector of this branch could then be replaced by an expression not containing it. The supposed basis would then contain more than the minimum number of vectors required and would therefore not be a basis.

A connected system of branches and nodes containing no cycle is, in topology, called a "tree." Thus, a basis for the incidence matrix of a connected network consists of the column vectors associated with a tree which contains every node of the network. An easy method is thereby provided both for securing a basis and for expressing all vectors in terms of it.

DETERMINING THE "EVALUATORS"

The "evaluators"—the ω vector[6]—may be determined in a manner similar to the way in which the "row-column numbers" are determined in distribution type models of linear programming.[7] In the model which is being developed ω_i is associated with the i^{th} equation. The value of any one such ω_i may be selected arbitrarily because the system of equations (3) has the same degree of linear dependence as the standard distribution model.[8] The other ω_k are then quickly determined since each basis branch consists of a node pair and the calculations are initiated at node 1 and proceed away from it to all the other nodes of the network along the appropriate basis branches (hence column vectors).

EXTREMAL PRINCIPLE AND THE FUNCTIONAL

In the case of flow problems to be considered here, it is assumed that to each branch j there corresponds a resistance function $r_j(q_j)$ such that

---

[6]Any solution of $\omega^T B = c^T$, where B is the basis at the current iteration. See [7], for example.

[7]In the following discussion some familiarity with linear programming is assumed. See, e.g., [3] and [5] for elaboration.

[8]This may be seen from the fact that the number of branches in a basis tree is one less than the number of nodes. In physical terms this corresponds to the fact that the voltage differences rather than absolute voltages are determined by the currents in a network.

$$R_j(q_j) = \int_0^{q_j} r_j(q)\,dq \tag{4}$$

is a convex function of q_j. For example, in the firewater system problem

$$r_j(q) = \alpha_j \operatorname{sgn}(q) \times |q|^k, \quad k > 1$$

and

$$R_j(q) = \frac{\alpha_j}{k+1} |q|^{k+1} \tag{5}$$

The convex programming problem, or extremal principle, characterizing the network flow is

$$\min. \sum_j R_j(q_j)$$

(6) subject to

$$\sum_j \epsilon_{ij} q_j = E_i, \quad i = 1, 2, \ldots$$

It may be established[9] that any optimal solution to this problem automatically satisfies the "head-loss," or Kirchhoff cycle conditions—viz, the algebraic sum of the head-losses around any closed loop consisting entirely of current carrying branches is zero. This principle thus achieves an essential simplification of the problem because in arriving at a solution it is only necessary to consider individual branches with the highly nonlinear loop conditions being transferred to the functional[10] in a much less recondite form.

APPROXIMATION OF THE FUNCTIONAL

To any desired degree of approximation each $R_j(q)$ may be replaced by a piecewise linear function so that the total, $\sum_j R_j(q_j)$, may also be approximated.[11] The resulting problem may then be replaced by an equivalent linear programming problem using the bounded variables techniques in order (a) to avoid proliferation of the number of vectors in the active basis and (b) to avoid increase in the number of structurally distinct vectors.[12]

---

[9] The authors wish to thank the referee for stimulating a sufficiently general proof of this (which will be published elsewhere) based on extending the Kuhn-Tucker theorem for this situation. He pointed out that Duffin, Birkhoff, and Diaz required $r_j(q)$ continuous and strictly increasing in their proofs.

[10] It should be noted that the objective here is to obtain a simulation and that this is accomplished by optimization with resulting gains in efficiency for the desired simulation. For mathematical proofs of this principle under heavier restrictions than are necessary, see [10] and [11].

[11] Additional research which eliminates the need for this approximation step will be reported in a subsequent paper. The method to be reported consists essentially of implicit refinement to the limit of the approximation process and of short-hand rules so that no approximation needs to be dealt with explicitly. This, of course, replaces the linear iterations with a nonlinear counterpart.

[12] See [4] for a general description of this method and others for approximating a nonlinear functional by a piecewise linear one.

In detail, let

$$q_j = q_j^+ - q_j^- ; \; q_j^+ , \; q_j^- \geq 0$$

and further let

(7)
$$q_j^+ = \sum_k \Delta_{jk}^+ , \; q_j^- = \sum_k \Delta_{jk}^-$$

$$0 \leq \Delta_{jk}^+ \leq \Delta_k^+$$

$$0 \leq \Delta_{jk}^- \leq \Delta_k^- .$$

Then, the approximating problem to (6) assumes the form[12]

$$\min. \sum_{j,k} \left[\rho_{jk}^+ \; \Delta_{jk}^+ + \rho_{jk}^- \; \Delta_{jk}^- \right]$$

subject to

(8)
$$\sum_j \epsilon_{ij} \left[\sum_k \left(\Delta_{jk}^+ - \Delta_{jk}^- \right) \right] = E_i$$

$$0 \leq \Delta_{jk}^+ \leq \Delta_k^+$$

$$0 \leq \Delta_{jk}^- \leq \Delta_k^-$$

where i, j, and k are each assigned their respective ranges.

METHODS OF COMPUTATION AND ILLUSTRATION

The method of computation on the network graph may be summarized as follows (the analogous steps on a tableau will be clear from the illustration). One starts with a solution of the node conditions admitting only branches which form a basis. The ω-vector is then calculated using the active basis, and from it, the Z_j is simply $\omega_h - \omega_t$, where node h is the "head" of branch j and node t is its "tail" as determined by its orientation. These are compared as usual with the c_j (in equation (8) the ρ_{jk}) to determine optimality or nonoptimality. If nonoptimal the vector to "come in" to the basis is expressed in terms of the active basis simply by going from the tail to the head of the come in branch on active basis branches. Those branches whose orientation is opposite to this direction of travel will have their "current" increased in the new solution, those with same orientation will have it decreased. In other words, the special structure of this problem is used to generate the information necessary to a simplex (or dual) procedure at each stage rather than using the general algorithm and storing additional information.

The example of Figure 1 may be used to provide a simple illustration for the computational routine to be employed. In this Figure an influx of 10 units is being put into the system at node 1 and effluxes of 2, 5, and 3 units, respectively, are being drawn from nodes 5, 7, and 9.

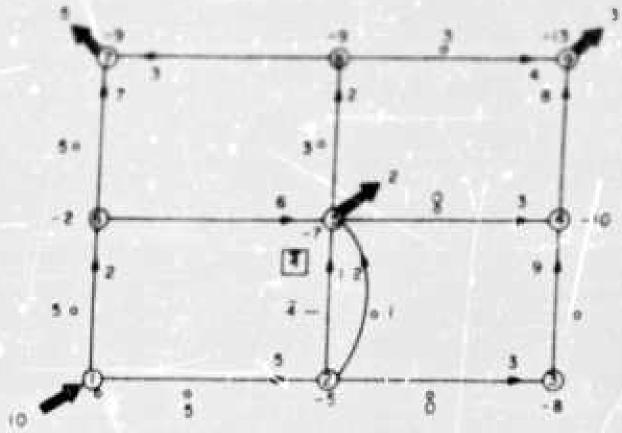

Figure 1 - Network Graph

The coefficients (here constants) for the variables which enter into the functional are shown inside the diagram opposite each arrowhead.[13] A physical analogy for these coefficients is found in the resistances, or head losses, over these branches in the circuit which are here assumed to be constants. The resistances are also assumed to be the same irrespective of the direction of flow.[14] Finally the branch 2, 5 is assumed to be characterized by a resistance of one up to 4 units of current and a resistance of two for greater currents. This can be visualized most easily by inserting two branches between nodes 2 and 5, designating the first as $\overline{2, 5}$, to indicate that it is limited in the amount of flow it will admit and the second by 2, 5 to indicate the absence of such an upper bound on the possible values of the flows through this (the latter) branch. The objective then is to characterize the current flows by minimizing the value of the functional overall branches of the network, taking account of the fact that $\overline{2, 5}$ is limited to a capacity of four units, as indicated by the value $\boxed{4}$ alongside its arrowhead, and that further flows along the branch will encounter the increased resistance associated with 2, 5.

Table 1 contains the corresponding incidence matrix with nodes entered in the stub and branches at the head. Each column is composed of the values ϵ_{ij} as described in section 2, above. The external influxes and effluxes for the system are noted at the right of the matrix in the column labelled "stipulations." Each row of the matrix then represents an equation which is a Kirchhoff node condition to describe the conservation of current. At the top of each branch is a value, c_j, to be entered into the functional and drawn from the amounts entered at the arrowheads on the branches of Figure 1.

Computation may proceed by means of either the Table or the Figure. For purposes of exposition, both devices will be used.

First an active basis is selected. To guarantee that a basis is, in fact, selected it is necessary, as previously described, only to choose branches in such a manner that all nodes in the system are attained without loops. Hence computations may commence by selecting a

[13] In the actual computations, it is advisable to utilize different colors to distinguish between these values, the programmed flows and the values for the ω vector.

[14] It would be possible to consider these resistances as different, depending on the direction of flow. Uni-directional flow could then be assured by blocking out counter flows with heavy resistance penalties.

TABLE 1
Incidence Matrix and Computing Routine

ω Components (basis set)	Functional coefficients	5	2	3	1	2	9	3	8	6	2	7	3	4	Stipulations
	Node \ Branch	1.2	1.6	2.3	$\overline{2.5}$	2.5	3.4	5.4	4.9	6.5	5.8	6.7	8.7	8.9	
0	1	1	1												= 10
-5	2	-1		1	1	1									= 0
-(5+3)	3			-1			1								= 0
-10 ~~-(5+3+9)~~	4						-1	-1	1						= 0
-7	5				-1	-1		1		-1	1				= -2
-2	6		-1							1		1			= 0
-9	7											-1	-1		= -5
-9	8										-1		1	1	= 0
-13	9								-1					-1	= -3
Basic indicators															
Program		$\dot{5}$	$\dot{5}$	$\mathring{0}$	$\bar{4}$	$\mathring{1}$	$\mathring{0}$				$\dot{3}$	$\dot{5}$		$\dot{3}$	
z values, nonbasis vectors					2			10	-4	5			0		
Basic indicators															
Program		$\dot{5}$	$\dot{5}$	$\mathring{0}$	$\bar{4}$	$\mathring{1}$		$\mathring{0}$			$\dot{3}$	$\dot{5}$		$\dot{3}$	
z values, nonbasis vectors					2		2		3	5			0		

promising basis set for the purpose of minimization. If this set is not optimal, a series of systematic iterations will secure such a set in a finite number of stages.

The basis thus initially chosen is indicated by the dots centered on the outside of the selected branches. It consists of the following branches: 1, 2; 1, 6; 2, 3; $\overline{2, 5}$; 3, 4; 5, 8; 6, 7; and 8, 9; as shown in Figure 1.[15] This set of dots is repeated directly below Table 1 in the first row labelled "Basis indicators." It designates the collection of column vectors which constitute an active basis. Immediately below these dots the row labelled "Program" shows the (unique) current flows which are readily ascertained from Figure 1 by reference to the branches in the basis and the influxes and effluxes.

Note that $\overline{2, 5}$ and 2, 5 are linearly dependent. Hence, at most, one of these vectors may enter into a basis. The other elements of a basis having been chosen, the total amount of current to flow between nodes 2 and 5 is uniquely determined. Since the objective is minimization, the optimal choice consists of filling up each such set of branches in the order of successively increasing c_j's. Thus, the flow of 5 between nodes 2 and 5 is allocated as follows: First as much as possible of this flow is allocated to $\overline{2, 5}$. Whenever such a link is used to capacity with overflow to one of its parallel links, such as 2, 5, it is not counted as part of an active basis.[16] For this reason $\overline{2, 5}$ receives a bar rather than a dot in the row of Table 1 labelled "Basis indicators," as well as alongside the program value $\bar{4}$ in Figure 1. The bar (replacing the dot) indicates that this branch is at capacity (with overflow) and is, therefore, not in the basis.

By virtue of the linear dependence of the Kirchhoff node conditions, exactly eight vectors here constitute a basis. The above method of choosing between $\overline{2, 5}$ and 2, 5 resolves the question of choice among links joining the same node pair, only one of which can be in an active basis.

To determine whether the programmed amounts are optimal, the methods of linear programming are now applied. Values, ω_i, are determined, one for each node of the network, as follows.[17] One such value may be chosen arbitrarily. Here the first such value, zero, is selected at node 1, and entered to the left of this node (under the ω components) in Table 1.[18] The remaining ω components are now uniquely determined by the condition that each pair of values ω_i and ω_k when multiplied by the corresponding elements of each basis vector and summed, must equal the value, c_j, located at the top of each such vector. Call the sum z_j which results from algebraically adding each pair of ω's so that the condition to be satisfied is $z_j = c_j$. Noting that zero appears as ω_1 opposite node 1 and that $c_{12} = 5$ it is apparent that $\omega_2 = -5$. Similarly, since $c_{16} = 2$ it follows that $\omega_6 = -2$ in order to obtain $\omega_1 - \omega_6 = 1 \times 0 + (-1)(-2) = z_{16} = c_{16} = 2$. Next, the value $\omega_3 = -8$ is determined from $\omega_2 - \omega_3 = z_{23} = c_{23} = 3$. Since $\omega_2 = -5$ it is evident that $\omega_3 = -(5+3)$, as shown, to achieve the desired equality.[19]

---

[15] The dot on branch 5, 4 should be ignored at this stage. It enters at a subsequent iteration when 3, 4 is eliminated, as will be explained subsequently.

[16] See [3] or [7].

[17] It may be helpful to regard these values as tentative node potentials in terms of which the flows are determined.

[18] Note that these values may also be determined directly from Figure 1. Thus, a value of zero is inserted at node 1. When c_{12} is subtracted from this amount, the value of ω_2 at node 2 is secured. Similarly, the value -8 for node 3 is obtained by subtracting c_{23} from ω_2, and so on.

[19] Note that these values may also be determined directly from Figure 1. Thus, a value of zero is inserted at node 1. When $c_{12} = 5$ is subtracted from this amount the value $\omega_2 = -5$ at node 2 is secured. Similarly the value $\omega_3 = -8$ for node 3 is obtained by subtracting c_{23} from ω_2, and so on.

Continuing in this fashion utilizing only vectors designated by a dot (and therefore in the basis) the remaining values in the left hand stub of Table 1 are thereby determined.[20]

All vectors in the active basis, therefore, necessarily have $z_j - c_j = 0$. If the program is optimal those vectors which are not in the active basis have values $|z_j| - c_j \leq 0$[21] and bounded variables, such as the value for 2, 5, will have values $z_j - c_j \geq 0$.

If any value is located which does not meet these conditions, it is necessary to iterate further. In the present instance $|z_{54}| > c_{54}$. This vector is therefore brought into the basis, as indicated by the upward pointing arrow just below the first row of "z values" of Table 1. To determine which vector is to be eliminated, recourse may be had to Figure 1. Since 5, 4 is being brought into the basis, the node at 4 can be secured via this route. That is, the vector 5, 4 may be stated in terms of the basis by tracing the unique path along the basis tree either from head to tail or tail to head. Thus, 5, 4 utilizes 5, 2; 2, 3; and 3, 4. The first is accorded a minus sign as the negative of 2, 5 and the latter are accorded positive signs. Only the latter (the ones accorded positive signs) are considered for purposes of designating the vector to be eliminated from the basis when 5, 4 is inserted. By recourse to standard procedures, including known methods of resolving degeneracy,[22] the vector 3, 4 is eliminated from the basis and 5, 4 inserted instead.

Crossing out the programmed value of zero along 3, 4 and inserting it instead at 5, 4, all other flows are left undisturbed at their previous values. Also, the only value of ω_i which requires alteration, is the one at node 4. Starting with $\omega_5 = -7$ and subtracting $c_{23} = 3$ from this amount, the new value $\omega_4 = -10$ is secured to replace the old one of -17, as shown in Figure 1.

The second row of dots in Table 1 indicates the new basis with the programmed amounts entered directly below the dots. To check for optimality, all vectors not in the active basis are evaluated as before. The test for optimality being successfully passed, as indicated by the numbers in the z value row at the bottom of the Table,[23] the problem is solved with the current flows as indicated in the "program" row.

The methods for convex programming over incidence matrices here developed, may be generalized to what may be called matrices of incidence type with nonzero entries other than ± 1. Specifically they may be considered as matrices for node conditions which generalize the Kirchhoff node conditions in that the current in a particular branch is to be multiplied by a number associated with the branch before entry into the node. Such extensions are important, for instance, in problems of accounting and financial analysis, and are encountered even in very simple cases such as the ones described in [6], the further refinement of these methods[24] to the exact functions and involving nonlinear iteration steps,[25] is also important for dealing with problems of risk and uncertainty. These topics will be treated in subsequent papers.[25]

[20] For a discussion of the theory underlying these procedures, see [3].

[21] Since flows may be in either direction, it is necessary either to use the absolute value of z_j or else to take account of the direction of flow. In links such as $\overline{2, 5}$, the direction of flow must be considered. The z_j which results from a negative flow, is simply the negative z_j for positive flow. Thus, it is unnecessary to carry two columns. In either case—positive or negative flow—on a capacitated link the true z_j must be greater than or equal to the corresponding c_j.

[22] See [2].

[23] I.e., $|z_j| \leq c_j$ for all nonbasis vectors and $z_j \geq c_j$ for $\overline{2, 5}$.

[24] See footnote 11, page 234, supra.

[25] E.g., A. Charnes and C. E. Lemke, "Continuous Limit Methods in Mathematical Programming," ONR Research Memo. No. 1, Systems Research Group, Northwestern University (forthcoming).

BIBLIOGRAPHY

[1] Babbit, H. E. and J. J. Doland, Water Supply Engineering (New York: McGraw-Hill Book Company, Inc., 1955).

[2] Charnes, A., "Degeneracy and Optimality in Linear Programming," Econometrica 20:2 (1952).

[3] ______ and W. W. Cooper, "Management Models and Industrial Applications of Linear Programming," Management Science, 4:1 (1957).

[4] ______, "Nonlinear Power of Adjacent Extreme Point Methods in Linear Programming," Econometrica 25:1 (1957).

[5] ______ and A. Henderson, An Introduction to Linear Programming (New York: John Wiley and Sons, Inc., 1953).

[6] ______ and M. Miller, "Programming and Financial Budgeting," Symposium on Techniques of Industrial Operations Research, June 1957, (Chicago: Illinois Institute of Technology, forthcoming).

[7] ______, "Dyadic Programming and Sub-dual Methods," ONR Research Memo. No. 21 (Lafayette; Purdue University, December 1957).

[8] ______ and C. E. Lemke, "Computational Theory of Linear Programming I: The Bounded Variables Problem," ONR Research Memo. No. 10 (Pittsburgh: Graduate School of Industrial Administration, Carnegie Institute of Technology, 1954).

[9] Prager, W., "Numerical Solution of the Generalized Transportation Problem" (Providence: Brown University, Department of Applied Mathematics, 1956) mimeo.

[10] Duffin, R. J., "Non-linear Networks," Bulletin of American Mathematical Society 52:833 (1946) and 53:963 (1947).

[11] Birkhoff, G. and Diaz, J. B., "Non-linear Network Problems," Quarterly of Applied Mathematics 13:431 (1956).

* * *